



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/737,374	12/16/2003	Roger Hansen	200312027-1	5369
22879 7590 09/22/2008 HEWLETT PACKARD COMPANY P O BOX 272400, 3404 E. HARMONY ROAD INTELLECTUAL PROPERTY ADMINISTRATION FORT COLLINS, CO 80527-2400				
EXAMINER				
TRUONG, LOAN				
ART UNIT		PAPER NUMBER		
2114				
NOTIFICATION DATE		DELIVERY MODE		
09/22/2008		ELECTRONIC		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM

mkraft@hp.com

ipa.mail@hp.com

### Office Action Summary

**Application No.**

10/737,374

**Applicant(s)**

HANSEN ET AL.

**Examiner**

LOAN TRUONG

**Art Unit**

2114

**Period for Reply** -- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 16 June 2008.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-16 and 18-37 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-16 and 18-37 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/5508)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_

**FINAL ACTION**

1. This office action is in response to the Request for continuation filed June 16, 2008 in application 10/737,374.
2. Claims 1-16 and 18-37 are presented for examination. Claims 18-19, 24 are amended. Claim 17 is previously cancelled.

***Response to Arguments***

3. Applicant's arguments filed June 16, 2008 have been fully considered but they are not persuasive.

In regard to claim 1-5, 7-13, 16 and 18-19 as being unpatentable over Chung et al., Pandya and Tsrigotis et al., applicant submits that the suggested combination is the result of the unallowable use of hindsight. Applicant argues that Chung teaches the use of non-byte addressable memory for storing checkpoint data, Pandya teaches TCP/IP processor and data processing that uses byte addressable RDMA memory but does not mention the use of meta-data and Tsrigotis creates a backup of cache memory on non-volatile memory that can include meta-data. Furthermore, applicant stated that those skilled in use of cache memory would be aware that the contents of cache are temporary and consist of frequently accessed information and therefore would not guarantee that the checkpoint information will be available when a primary process fails. A further look into the prior art reveals that no where in Chung stated a "non-byte addressable memory" as mention by applicant. Secondly, applicant agrees that Pandya teaches the limitation of a data processing engine using byte addressable RDMA memory but does not

mention the use of meta-data or storing data for backup purposes. Examiner would like to clarify that the limitation of storing data for backup purposes is taught by Chung while the limitation of using meta-data is taught by Tsrigotis. Finally, Tsrigotis cache system describes a structure to store cache information in which a primary copy of cached information is stored may be called a primary buffer (*col. 5 lines 19-23*) and where in one embodiment, the primary buffer are implemented as cyclic buffers that may be stored in persistent memory (*col. 5 lines 55-61*). Tsrigotis not only taught the limitation of using meta-data but also resolved applicants' raised arguments of the cache memory being temporary with the primary buffer implemented in persistent memory.

In regard to claim 18, applicant stated that Tsrigotis does not teach the limitation of “establishing a connection to a process requesting access to the checkpoint data; and binding the access information to the connection” but acknowledged that Tsrigotis teach HTTP protocol to handle messages passing between a client and server. According to Wikipedia, typically an HTTP client initiates a request and establishes a Transmission Control Protocol (TCP) connection to a particular port, port 80 by default, on a host ([http://en.wikipedia.org/wiki/Http\\_protocol](http://en.wikipedia.org/wiki/Http_protocol)). For this reason, examiner interpret that Tsrigotis does teach the above limitation.

In regard to claim 6, refer to claim 1 above.

In regard to claim 14-15, refer to claim 10 above.

In regard to claim 20, same reason as claim 10. Furthermore, applicant argues that Ho does not teach the limitation of "authenticating a persistent memory manager during initialization of address protection and translation tables on the persistent memory unit," but only taught a software protection program that includes a signature engraving program, a signature authentication program, and an engraved signature. Examiner disagree, because Ho taught the software protection program for the purpose of preventing unauthorized use (*paragraph 0003*), by having the software protection program checks the signature during system startup for authentication purposes (*paragraph 0015*).

In regard to claim 21-23 and 25, applicant stated that Pandya does not teach the limitation of "authenticate requests from remote processors, and provide access information to authenticated remote processors based on address protection and translation tables in the persistent memory unit, the access information includes meta-data regarding the contents and layout of memory regions within the persistent memory unit." Examiner would like to point out that the latter part of the argued limitation "*the access information includes meta-data regarding the contents and layout of memory regions within the persistent memory unit*" is not presented in any one of claims 21-23 or 25. Applicant acknowledged that Pandya teaches the host application register a memory region that it wishes to use in RDMA transactions and communicates this information to a peer application on a remote end but did not considered Pandya's RDMA commands which could utilized protection key look-up and address translation

as per the RDMA initialization (*paragraph 0124 lines 115-118*). Claims 22-25 depends on rejected claim 21 and therefore are also rejected. Refer to rejections below.

In regard to claims 26, 32 and 35, applicant stated that the cited references do not teach the limitation of "receiving access information to physical addresses of checkpoint data in the persistent memory from the persistent memory unit." Applicant argued that Pandya teach the host application and not the RDMA engine provides information regarding the RDMA memory region to a peer application. Examiner disagree due to the fact that Pandya teach the limitation of the RDMA engine provides various capabilities necessary for enabling remote direct memory access where it has tables with information like RDMA region and the access keys and virtual address translation functionality and where the RDMA performs the data transfer and interprets the received RDMA commands (*paragraph 0124 lines 14-19*). For this reason, even if the host is transmitting the access information, RDMA engine provides the capabilities necessary to do so and therefore satisfy the limitation of "receiving access information from the persistent memory unit." Applicant also argued that Tsirigotis has no reason to provide information regarding the cache memory to another host application because the cache memory is only useful to a particular host application. Refer to arguments for Tsirigotis cache memory above. Claims 27-31, 33-34 and 36-37 depends on rejected claims 26, 32 and 35 and therefore are also rejected. Refer to rejections below.

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
  2. Ascertaining the differences between the prior art and the claims at issue.
  3. Resolving the level of ordinary skill in the pertinent art.
  4. Considering objective evidence present in the application indicating obviousness or nonobviousness.
4. Claims 1-5, 7-13, 16, 18-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chung et al. (US 6,195,760) in further view of Pandya (US 2004/0037319) in further view of Tsigotis et al. (US 6,883,068).

In regard to claim 1, Chung et al. teach a system for storing checkpoint state information, comprising:

A network interface to an external network (*network, fig. 1, 100, col. 3 lines 60-62*); and  
a persistent memory unit (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44*) coupled to the network interface (*connected to the network, fig. 1, 100, col. 4 lines 41-44*), wherein:

the persistent memory unit (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44*) is configured to receive the checkpoint data (*periodically receives from each fault-protected application module running on the network the most current state of that application, col. 4 lines 41-44*) via a memory write command from a primary process (*hot backup where each copies of an application can process client request and states are synchronized among multiple copies, col. 2 lines 7-14*) through the network interface (*connected to the network, fig. 1, 100, col. 4 lines 41-44*), and to provide access to the checkpoint data via a memory read command from the backup process (*last operating state provided to the backup, col. 4 lines 34-40*), through the network interface (*last stored state is retrieved from the memory of Checkpoint Server connected to network, fig. 1, 110, 100, col. 4 lines 41-48*)); and

The backup process provides recovery capability in the event of a failure of the primary process (*idle or backup application module assume the functioning of a failed primary application module upon failure-detection, col. 4 lines 34-40*).

Chung et al. does not explicitly teach the system for storing checkpoint state information comprising of a remote direct memory read and write command and the persistent memory unit is configured with byte-level memory access granularity.

Pandya teaches the TCP/IP processor and engine using RDMA implementing a RDMA engine that provides various capabilities necessary for enabling remote direct memory access (*paragraph 0124 lines 13-15*) where the



RDMA transfer may include read and write operation from a region with a certain number of bytes with a specific offset (*paragraph 0124 lines 80-86*).

It would have been obvious to modify the system of Chung et al. by adding Pandya TCP/IP processor and engine. A person of ordinary skill in the art at the time of applicant's invention would have been motivated to make the modification because it would provide a new high performance and low latency to relieve the host processor of the server performance impact (*paragraph 0067*).

Chung et al. and Pandya does not teach meta-data regarding the contents and layout of memory regions with the persistent memory.

Tsirigotis et al. teach a persistent cache including metadata that contains information about how data is stored and formatted within persistent memory and may be stored at the same location on each disk (*col. 4 lines 22-30*).

It would have been obvious to modify the system of Chung et al. and Pandya by adding Tsirigotis et al. apparatus for implementing a cache replacement scheme. A person of ordinary skill in the art at the time of applicant's invention would have been motivated to make the modification because it would provide information for accessing indexes corresponding to buffers within the persistent storage (*col. 4 lines 22-30*).

In regard to claim 2, Chung et al. teach the system of claim 1, further comprising:

A persistent memory manager (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44*) configured to provide address context information to the network interface (*pathname location of each copy of the application module on the host computer, fig. 2, 200*).

Chung et al. and Pandya does not teach the system comprising a persistent manager keep the meta-data on the persistent memory unit consistent with the checkpoint data stored on the persistent memory unit.

Tsirigotis et al. teach a persistent cache including metadata that contains information about how data is stored and formatted within persistent memory and may be stored at the same location on each disk (*col. 4 lines 22-30*).

Refer to claim 1 for motivational statement.

In regard to claim 3, Chung et al. teach the system of claim 1, wherein the persistent memory unit (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44*) is configured to provide access to the checkpoint data to another processor (*Checkpoint server transmit last stored state to new primary application module, fig. 1, 110, H1-6, col. 4 lines 41-48*), and the backup process is executed by the other processor (*backup for A application on multiple host computer, fig. 2, H2 and H3*).

Chung et al. does not explicitly teach providing a remote direct memory read access.

Pandya teaches the TCP/IP processor and engine using RDMA implementing a RDMA engine that provides various capabilities necessary for enabling remote direct memory access (*paragraph 0124 lines 13-15*) where the RDMA transfer may include

read and write operation from a region with a certain number of bytes with a specific offset (*paragraph 0124 lines 80-86*).

Refer to claim 1 for motivational statement.

In regard to claim 4, Chung et al. teach the system of claim 1, wherein the persistent memory unit (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44*) provides the checkpoint data by the backup process after the primary process fails (*Checkpoint server transmit last stored state to new primary application module, fig. 1, 110, H1-6, col. 4 lines 41-48*).

Chung et al. does not explicitly teach providing data through remote direct memory read access.

Pandya teaches the TCP/IP processor and engine using RDMA implementing a RDMA engine that provides various capabilities necessary for enabling remote direct memory access (*paragraph 0124 lines 13-15*) where the RDMA transfer may include read and write operation from a region with a certain number of bytes with a specific offset (*paragraph 0124 lines 80-86*).

Refer to claim 1 for motivational statement.

In regard to claim 5, Chung et al. teach the system of claim 1, wherein the persistent memory unit (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44*) is configured to store multiple sets of checkpoint data sent from the processor at successive time intervals (*checkpoint technique to periodically take snapshots of the running state in a stable storage media, col. 1 lines 49-58*).

Chung et al. does not explicitly teach the system for storing data through remote direct memory writes.

Pandya teaches the TCP/IP processor and engine using RDMA implementing a RDMA engine that provides various capabilities necessary for enabling remote direct memory access (*paragraph 0124 lines 13-15*) where the RDMA transfer may include read and write operation from a region with a certain number of bytes with a specific offset (*paragraph 0124 lines 80-86*).

Refer to claim 1 for motivational statement.

In regard to claim 7, Chung et al. teach the system of claim 1, wherein the primary process (*primary for application A, fig. 2, 202*) writes the checkpoint data (*snapshot of the running state of the primary application, col. 1 lines 49-58*) to the persistent memory unit (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44*) independently from the backup process (*backup for application A, fig. 2, 203-205*).

Chung et al. does not explicitly teach the system for remote direct memory writes.

Pandya teaches the TCP/IP processor and engine using RDMA implementing a RDMA engine that provides various capabilities necessary for enabling remote direct memory access (*paragraph 0124 lines 13-15*) where the RDMA transfer may include read and write operation from a region with a certain number of bytes with a specific offset (*paragraph 0124 lines 80-86*).

Refer to claim 1 for motivational statement.

In regard to claim 8, Chung et al. teach the system of claim 1, wherein the persistent memory unit (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44*) is configured as part of a memory access-enabled system area network (*Checkpoint Server is connected to network, fig. 1, 110, 100*).

Chung et al. does not explicitly teach the system of claim 1, is configured as part of a remote direct memory access.

Pandya teaches the TCP/IP processor and engine using RDMA implementing a RDMA engine that provides various capabilities necessary for enabling remote direct memory access (*paragraph 0124 lines 13-15*) where the RDMA transfer may include read and write operation from a region with a certain number of bytes with a specific offset (*paragraph 0124 lines 80-86*).

Refer to claim 1 for motivational statement.

In regard to claim 9, Chung et al. teach the system of claim 1, wherein the persistent memory unit to provide access information to authenticated remote processors (*application module register itself for its own failure and recovery process, col. 3 lines 1-15*).

Chung et al. does not explicitly teach the system where the persistent memory unit is configured with address protection and translation tables to authenticate requests from remote processors.

Pandya teaches the TCP/IP processor and engine using RDMA implementing a RDMA engine that provides various capabilities necessary for enabling remote direct memory access which has the RDMA look-up table that include information like RDMA

region and the access keys, and virtual address translation functionality (*paragraph 0124 lines 13-18*)

Refer to claim 1 for motivational statement.

In regard to claim 10, Chung et al. teach a method for recovering the operational state of a primary process, comprising:

A persistent memory unit (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44*); writing checkpoint data regarding the operational state of the primary process (*periodically receives from each fault-protected application module running on the network the most current state of that application, col. 4 lines 41-44*) in the persistent memory unit (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44*); and

Reading the checkpoint data from the persistent memory unit via a memory read from the backup process (*last operating state of the failed application module must be provided to the backup application module, col. 4 lines 34-40*).

Chung et al. does not teach the method of mapping virtual addresses of a persistent memory unit to physical addresses of the persistent memory unit where the persistent memory unit is addressable at byte-level granularity and remote direct memory read and write.

Pandya teaches the TCP/IP processor and engine using RDMA implementing a RDMA engine that provides various capabilities necessary for enabling remote direct memory access (*paragraph 0124 lines 13-15*) where the RDMA transfer may include read and write operation from a region with a certain number of bytes with a specific

offset (*paragraph 0124 lines 80-86*). The RDMA engine also provide tables like RDMA look-up tables that include information like RDMA region and the access keys, and virtual address translation functionality (*paragraph 0124 lines 13-17*).

Refer to claim 1 for motivational statement.

Chung et al. and Pandya does not teach a method for storing access information to the physical addresses of the checkpoint data in the persistent memory unit when the primary process opens a memory region for the checkpoint data; and providing the access information to subsequent requestors of the checkpoint data.

Tsirigotis et al. teach the method for implementing a cache replacement scheme by implementing a hash table entry comprises an object identifier corresponding to the object, an index corresponding to a position within a buffer and if the object is also stored in main memory, a corresponding hash entry may also reference a memory address corresponding to an address in memory where the object is stored (*col. 9 lines 10-43, fig. 6*). Tsirigotis et al. also teach of receiving a request and determined whether an object is and return the object to the client (*col. 5 lines 40-61, fig. 2*).

Refer to claim 1 for motivational statement.

In regard to claim 11, Chung et al. teach the method of claim 10, further comprising:  
Providing context information regarding the addresses to the primary process and the backup process.

*It is inherent that the network of fig. 1 with the plurality of host computer H1-H6 if connected in an Ethernet network would have their own IP address to distinct one host*

*computer from another (col. 3 lines 60-67). Furthermore, the registration request from each failure-protected application module included a list of the host computers on which the application modules resided and where on each the executable program can be found (col. 4 lines 49-60).*

In regard to claim 12, Chung et al. teach the method of claim 10, further comprising:  
providing the checkpoint data to the backup process upon failure of the primary process  
*(Checkpoint server transmit last stored state to new primary application module, fig. 1, 110, H1-6, col. 4 lines 41-48).*

Chung et al. does not teach remote direct memory reads.

Pandya teaches the TCP/IP processor and engine using RDMA implementing a RDMA engine that provides various capabilities necessary for enabling remote direct memory access (*paragraph 0124 lines 13-15*) where the RDMA transfer may include read and write operation from a region with a certain number of bytes with a specific offset (*paragraph 0124 lines 80-86*).

Refer to claim 1 for motivational statement.

In regard to claim 13, Chung et al. teach the method of claim 10, further comprising:  
Overwriting the checkpoint data with current checkpoint data (*when a failure of the primary application, the checkpoint data of the last stored state of the failed primary is supplied to the backup application module, col. 1 lines 49-57*).



In regard to claim 16, Chung et al. teach the method of claim 10, further comprising:

Allowing the backup process to read previously unread portions of the checkpoint data upon failure of the primary process (*Checkpoint server transmit last stored state to new primary application module, fig. 1, 110, H1-6, col. 4 lines 41-48*); and

Resuming functions performed by the primary process with the backup process (*checkpoint data of the last stored state of the failed primary application module is supplied to the backup application module, col. 1 lines 52-58*).

Chung et al. does not teach the method of remote direct memory read.

Pandya teaches the TCP/IP processor and engine using RDMA implementing a RDMA engine that provides various capabilities necessary for enabling remote direct memory access (*paragraph 0124 lines 13-15*) where the RDMA transfer may include read and write operation from a region with a certain number of bytes with a specific offset (*paragraph 0124 lines 80-86*).

Refer to claim 1 for motivational statement.

In regard to claim 18 Chung et al. and Pandya does not explicitly disclosed the method of claim 10, further comprising: establishing a connection to a process requesting access to the checkpoint data; and binding the access information to the connection.

Tsirigotis et al. teach the method for implementing a cache replacement scheme by implementing protocol that defines how a client and an origin server communicate.

The request may be retrieve operation which causes the origin server to retrieve the object identified by the URL and transmit it to the requesting client (*col. 1 lines 34-48*).

Refer to claim 1 for motivational statement.

In regard to claim 19, Chung et al. does not teach the method of claim 10, further comprising: verifying authentication information from the subsequent requestors.

Pandya teach the method for TCP/IP processor and engine using RDMA by implementing agreement on doing the RDMA transfers and allocating memory regions and permissions through access control keys that get shared (*paragraph 0124 lines 91-96*).

Refer to claim 1 for motivational statement.

\*\*\*\*\*

5. Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over Chung et al. (US 6,195,760) in further view of Pandya (US 2004/0037319) in further view of Tsirigotis et al. (US 6,883,068) in further view of Wang (US 7,082,553).

In regard to claim 6, Chung et al. does not explicitly teach the system of claim 5, wherein the persistent memory unit provides the multiple sets of checkpoint data through remote direct memory reads upon request by the backup process at one time.

Pandya teaches the TCP/IP processor and engine using RDMA implementing a RDMA engine that provides various capabilities necessary for enabling remote direct

memory access (*paragraph 0124 lines 13-15*) where the RDMA transfer may include read and write operation from a region with a certain number of bytes with a specific offset (*paragraph 0124 lines 80-86*).

Refer to claim 1 for motivational statement.

Chung et al., Pandya and Tsirigotis et al. does not teach the system wherein the persistent memory unit provides the multiple sets of checkpoint data upon request by the backup process at one time.

Wang teaches the for providing reliability and availability in a distributed component object model object oriented system by implementing checkpoint request to store current checkpoint of each object in registry image of disk storage (*fig. 8, col. 9 lines 21-30*).

It would have been obvious to modify the method of Chung et al., Pandya and Tsirigotis et al. by adding Wang distributed component object model object oriented system. A person of ordinary skill in the art at the time of applicant's invention would have been motivated to make the modification because it would help to quickly recover from a failure (*col. 9 lines 30-46*).

\*\*\*\*\*

6. Claims 14-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chung et al. (US 6,195,760) in further view of Pandya (US 2004/0037319) in further view of Tsirigotis et al. (US 6,883,068) in further view of St. Pierre et al. (US 6,141,773).

In regard to claim 14, Chung et al., Pandya and Tsigotis et al. does not teach the method of claim 10, further comprising: appending updated checkpoint data to at least one previous set of the checkpoint data.

St. Pierre et al. disclosed the method of backing up and restoring data in a computer storage system where differential backup is formed by the identified changed segments omitting at least on the segments that has not been changed (*col. 5 lines 30-63*). A differential bit file may be constructed including copies of only the changed data segments (*fig. 13, 111a-111d*). The differential bit file captures changes to a logical entity as contiguous (*col. 17 lines 28-38*).

It would have been obvious to modify the method of Chung et al., Pandya and Tsigotis et al. by adding St. Pierre et al. method of backing up data in a computer storage system (*col. 5 lines 30-63*). A person of ordinary skill in the art at the time of applicant's invention would have been motivated to make the modification because it would permits recovery from errors, including use of a mirror for data located at a remote facility that also permits recoveries from catastrophic failure (*col. 5 lines 25-28*).

In regard to claim 15, Chung et al. teach the method of claim 10, further comprising: clearing the portion of the multiple sets of checkpoint data (*periodically receives state updates from the primary application module, col. 1 lines 64-67*).

\*\*\*\*\*

7. Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over Chung et al. (US 6,195,760) in further view of Pandya (US 2004/0037319) in further view of Tsirigotis et al. (US 6,883,068) in further view of Ho et al. (US 2002/0073325).

In regard to claim 20, Chung et al. disclosed the method of claim 10, further comprising:

a persistent memory manager (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44*) during address protection and translation tables (*table, fig. 2, 200, col. 7 lines 1-5*) on the persistent memory unit (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44*).

Chung et al., Pandya and Tsirigotis et al. does not explicitly teach the method of authenticating a persistent memory manager during initialization.

Ho et al. teach the method of authenticating software licenses by implementing a persistent storage medium comprising a signature authentication program in the software protection program (*paragraph 0023*).

It would have been obvious to modify the system of Chung et al., Pandya and Tsirigotis et al. by adding Ho et al. method of authenticating software licenses. A person of ordinary skill in the art at the time of applicant's invention would have been motivated to make the modification because it would provide protection and allow legitimate backup copies (*paragraph 0011*).

\*\*\*\*\*

8. Claims 21-23 and 25-28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chung et al. (US 6,195,760) in further view of Stiffer et al. (US 6,622,263) in further view of Pandya (US 2004/0037319).

In regard to claim 21, Chung et al. disclosed a computer product, comprising:

Computer executable instructions (*application modules, col. 4 lines 1-15*) operable to: allow memory access from a remote processor via a network (*Checkpoint Server connected to the network periodically receives the most current state of each fault-protected application module stored in its memory, col. 4 lines 40-45*), wherein the memory access references a persistent memory (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44*);

Chung et al. does not teach a computer product reference to a persistent memory virtual address; store checkpoint data from a primary process; translate the virtual address to a physical address in the persistent memory unit wherein the persistent memory unit is addressable at byte-level granularity; and allow access to the checkpoint data for use in a backup process.

Stiffer et al. disclosed the apparatus for achieving system-directed checkpointing without specialized hardware assistance with the implementation of a memory map to convert virtual addresses used by application and by the operating system into physical addresses that point to specific locations in main memory. Each memory-map entry, in addition to containing virtual to physical address translation information, contains other information as well (*col. 5 lines 5-14*). Stiffer et al. also disclosed the method where each computer other than the standby sends checkpoint data to its neighbor on the right

and each stores in its shadow memory (*col. 14 lines 40-46*), when a fault is detected, the computer to the right takes over the tasks that was executing prior to the fault (*col. 14 lines 47-55*).

It would have been obvious to modify the system of Chung et al. by adding Stiffler et al. apparatus for achieving system-directed checkpointing without specialized hardware assistance. A person of ordinary skill in the art at the time of applicant's invention would have been motivated to make the modification because it would establishing and recording a consistent system state from which all running applications can be safely resumed following a fault (*col. 1 lines 13-17*).

Chung et al. and Stiffler et al. does not explicitly teach a computer product comprising a remote direct memory access and authenticate requests from remote processors, and provide access information to authenticated remote processors based on address protection and translation tables in the persistent memory unit;

Pandya teaches the TCP/IP processor and engine using RDMA implementing a RDMA engine that provides various capabilities necessary for enabling remote direct memory access (*paragraph 0124 lines 13-15*) where the RDMA transfer may include read and write operation from a region with a certain number of bytes with a specific offset (*paragraph 0124 lines 80-86*). The RDMA engine also provided tables that include information like RDMA region and the access keys, and virtual address translation functionality (*paragraph 0124 lines 15-20*). The RDMA commands may also go through protection key look-up and address translation as per the RDMA initialization (*paragraph 0124 lines 115-118*).

Refer to claim 1 for motivational statement.

In regard to claim 22, Chung et al. does not explicitly teach the computer product of claim 21, further comprising: computer executable instructions operable to: allow the processor to access address context information.

Stiffer et al. teach the method of checkpointing and fault recover software runs on standard platforms to detect observable malfunctions such as out-of-range address or exceeding a allocated range defined for a given data structure (*col. 4 lines 24-36*).

Refer to claim 21 for motivational statement.

In regard to claim 23, Chung et al. teach the computer product of claim 21, further comprising: computer executable instructions operable to:

Store multiple updates to the checkpoint data sent at successive time intervals (*checkpoint technique to periodically take snapshots of the running state in a stable storage media, col. 1 lines 49-58*).

In regard to claim 25, Chung et al. disclosed the computer product of claim 21, wherein the persistent memory (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44*) is configured as part of a memory access-enabled system area network (*Checkpoint Server is connected to network, fig. 1, 110, 100*).

Chung et al. and Stiffer et al. does not explicitly teach the system of claim 21, is configured as part of a remote direct memory access.



Pandya teaches the TCP/IP processor and engine using RDMA implementing a RDMA engine that provides various capabilities necessary for enabling remote direct memory access (*paragraph 0124 lines 13-15*) where the RDMA transfer may include read and write operation from a region with a certain number of bytes with a specific offset (*paragraph 0124 lines 80-86*).

Refer to claim 1 for motivational statement.

In regard to claim 26, Chung et al. teach an apparatus comprising:

means for communicatively coupling (*configuring a fail-over process, col. 1 lines 36-37*) a persistent memory unit (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44*) to a network (*network, fig. 1, 110, col. 4 lines 41-44*) that enables access to the memory unit (*A checkpoint Server connected to network periodically receives from each fault-protected application module running on the network, col. 4 lines 41-44*);

means for receiving the checkpoint data for a primary process (*periodically receives from each fault-protected application module running on the network the most current state of that application, col. 4 lines 41-44*) in the persistent memory unit (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44*) via the network (*network, fig. 1, 110, col. 4 lines 41-44*); and

Means for allowing the primary process to access the checkpoint data (*last operating state of the failed application module must be provided to the backup application module, col. 4 lines 34-40*) via the network (*Checkpoint Server connected to network to receives from each fault-protected*).

Chung et al. does not explicitly teach the means for mapping virtual addresses of the persistent memory unit to physical addresses of the persistent memory unit; means for allowing a backup process to access the checkpoint data via the network.

Stiffler et al. disclosed the method of system-directed checkpointing without specialized hardware assistance with a memory map of virtual addresses to physical addresses (*col. 5 lines 2-8*). et al. also disclosed the method where each computer other than the standby sends checkpoint data to its neighbor on the right and each stores in its shadow memory (*col. 14 lines 40-46*), when a fault is detected, the computer to the right takes over the tasks that was executing prior to the fault (*col. 14 lines 47-55*).

It would have been obvious to modify the system of Chung et al. by adding Stiffler et al. apparatus for achieving system-directed checkpointing without specialized hardware assistance. A person of ordinary skill in the art at the time of applicant's invention would have been motivated to make the modification because it would establishing and recording a consistent system state from which all running applications can be safely resumed following a fault (*col. 1 lines 13-17*).

Chung et al. and Stiffer et al. does not explicitly teach an apparatus comprising of means to enables remote direct memory read write access to a persistent memory, wherein the persistent memory unit is addressable at byte-level granularity; means for receiving access information to physical addresses of checkpoint data in the persistent memory from the persistent memory unit.

Pandya teaches the TCP/IP processor and engine using RDMA implementing a RDMA engine that provides various capabilities necessary for enabling remote direct

memory access (*paragraph 0124 lines 13-15*) where the RDMA transfer may include read and write operation from a region with a certain number of bytes with a specific offset (*paragraph 0124 lines 80-86*). The RDMA engine also provided tables that include information like RDMA region and the access keys, and virtual address translation functionality (*paragraph 0124 lines 15-20*). The RDMA commands may also go through protection key look-up and address translation as per the RDMA initialization (*paragraph 0124 lines 115-118*).

Refer to claim 1 for motivational statement.

In regard to claim 27, Chung et al. does not explicitly teach the apparatus of claim 26, further comprising: means for allowing the primary process and the backup process to access context information regarding the addresses.

Stiffer et al. teach the method of checkpointing and fault recover software runs on standard platforms to detect observable malfunctions such as out-of-range address or exceeding a allocated range defined for a given data structure (*col. 4 lines 24-36*).

Refer to claim 26 for motivational statement.

In regard to claim 28, Chung et al. teach the apparatus of claim 26, further comprising: means for allowing the backup process to access the checkpoint data upon failure of the primary process (*Checkpoint server transmit last stored state to new primary application module, fig. 1, 110, H1-6, col. 4 lines 41-48*).

\*\*\*\*\*

9. Claim 24 is rejected under 35 U.S.C. 103(a) as being unpatentable over Chung et al. (US 6,195,760) in further view of Stiffer et al. (US 6,622,263) in further view of Pandya (US 2004/0037319) in further view of Wang (US 7,082,553).

In regard to claim 24, Chung et al. and Stiffer et al. Pandya does not explicitly teach the computer product of claim 21, further comprising: computer executable instructions operable to: allow the backup process to access the multiple sets of the checkpoint data at one time.

Wang teaches the for providing reliability and availability in a distributed component object model object oriented system by implementing checkpoint request to store current checkpoint of each object in registry image of disk storage (*fig. 8, col. 9 lines 21-30*).

It would have been obvious to modify the method of Chung et al. and Stiffer et al. and Pandya by adding Wang distributed component object model object oriented system. A person of ordinary skill in the art at the time of applicant's invention would have been motivated to make the modification because it would help to quickly recover from a failure (*col. 9 lines 30-46*).

\*\*\*\*\*

10. Claims 29-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chung et al. (US 6,195,760) and in further view of Stiffer et al. (US 6,622,263) in further view of Pandya (US 2004/0037319) in further view of St. Pierre et al. (US 6,141,773).

In regard to claim 29, Chung et al. teach the apparatus of claim 26, further comprising: means for overwriting the checkpoint data with current checkpoint data (*when a failure of the primary application, the checkpoint data of the last stored state of the failed primary is supplied to the backup application module, col. 1 lines 49-57*).

Chung et al. and Stiffler et al. and Pandya does not explicitly teach the means for creating multiple sets of checkpoint data by appending updated checkpoint data to at least one previous set of the checkpoint data;

St. Pierre et al. disclosed the method of backing up and restoring data in a computer storage system where differential backup is formed by the identified changed segments omitting at least on the segments that has not been changed (*col. 5 lines 30-63*). A differential bit file may be constructed including copies of only the changed data segments (*fig. 13, 111a-111d*). The differential bit file captures changes to a logical entity as contiguous (*col. 17 lines 28-38*).

Refer to claim 14 for motivational statement.

In regard to claim 30, Chung et al. disclosed the apparatus of claim 29, further comprising: means for periodically supplying (*Checkpoint server periodically receives from each fault-protected application module running on the network, fig. 1, col. 4 lines 41-44*) at least a portion of the multiple sets of checkpoint data (*snapshot of the running state of the primary application, col. 1 lines 49-58*) in the backup process (*Replica-Manager stores information necessary to effect recovery of an entire host computer running several different application*

*modules, fig. 2, 200, col. 5 lines 21-30).*

In regard to claim 31, Chung et al. disclosed the apparatus of claim 30, further comprising:  
means for providing previously unread portions of the checkpoint data to the backup process  
upon failure of the primary process (*upon failure detection of an application module, the last  
stored state of the failed application is retrieved from the memory of Checkpoint Server, col.  
4 lines 45-48).*

\*\*\*\*\*

11. Claims 32-33 and 35-37 are rejected under 35 U.S.C. 103(a) as being unpatentable over  
Chung et al. (US 6,195,760) in further view of Pandya (US 2004/0037319).

In regard to claim 32, Chung et al. teach a method for recording the operational state of a  
primary process, comprising: checkpoint data (*periodically receives from each fault-protected  
application module running on the network the most current state of that application, col. 4 lines  
41-44*) regarding the operational state of the primary process (*WatchDog periodically polls  
application module and reports any failure to the ReplicaManager, col. 7 lines 42-49*) in a  
memory unit (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44*) via a memory access write  
command (*last stored state is retrieved from the memory of Checkpoint Server connected to  
network, fig. 1, 110, 100, col. 4 lines 41-48).*

Chung et al. does not explicitly teach the method for receiving access information  
to physical addresses of checkpoint data in the persistent memory from the persistent

memory unit; and accessing checkpoint data in persistent memory via a remote direct memory access wherein the persistent memory unit is addressable at byte-level granularity.

Pandya teaches the TCP/IP processor and engine using RDMA implementing a RDMA engine that provides various capabilities necessary for enabling remote direct memory access (*paragraph 0124 lines 13-15*) where the RDMA transfer may include read and write operation from a region with a certain number of bytes with a specific offset (*paragraph 0124 lines 80-86*). The RDMA engine also provided tables that include information like RDMA region and the access keys, and virtual address translation functionality (*paragraph 0124 lines 15-20*). The RDMA commands may also go through protection key look-up and address translation as per the RDMA initialization (*paragraph 0124 lines 115-118*).

Refer to claim 1 for motivational statement.

In regard to claim 33, Chung et al. teach the method of claim 32, further comprising: overwriting the checkpoint data in the persistent memory unit with current checkpoint data via a direct memory access write command (*when a failure of the primary application, the checkpoint data of the last stored state of the failed primary is supplied to the backup application module, col. 1 lines 49-57*).

Chung et al. does not explicitly teach the method for transmitting data in persistent memory via a remote direct memory access.

Pandya teaches the TCP/IP processor and engine using RDMA implementing a RDMA engine that provides various capabilities necessary for enabling remote direct memory access (*paragraph 0124 lines 13-15*) where the RDMA transfer may include read and write operation from a region with a certain number of bytes with a specific offset (*paragraph 0124 lines 80-86*).

Refer to claim 1 for motivational statement.

In regard to claim 35, Chung et al. teach a method for retrieving the operational state of a primary process, comprising: transmitting a memory access read command via network to a memory unit from a backup process for the primary process (*the last stored state of that failed application module is retrieved from the memory of Checkpoint Server and provided to the new primary application module for continued processing, col. 4 lines 44-48*).

Chung et al. does not explicitly teach the method for receiving access information to physical addresses of checkpoint data in the persistent memory from the persistent memory unit; and transmitting a remote direct memory access read wherein the persistent memory unit is addressable at byte-level granularity.

Pandya teaches the TCP/IP processor and engine using RDMA implementing a RDMA engine that provides various capabilities necessary for enabling remote direct memory access (*paragraph 0124 lines 13-15*) where the RDMA transfer may include read and write operation from a region with a certain number of bytes with a specific offset (*paragraph 0124 lines 80-86*). The RDMA engine also provided tables that include information like RDMA region and the access keys, and virtual address



translation functionality (*paragraph 0124 lines 15-20*). The RDMA commands may also go through protection key look-up and address translation as per the RDMA initialization (*paragraph 0124 lines 115-118*).

Refer to claim 1 for motivational statement.

In regard to claim 36, Chung et al. teach the method of claim 35, further comprising: periodically transmitting the memory access read command to retrieve at least a portion of the checkpoint data for the backup process (*Checkpoint Server periodically receives from each fault-protected application module running on the network the most current state of that application, col. 4 lines 41-44*).

Chung et al. does not explicitly teach the method for transmitting the remote direct memory access.

Pandya teaches the TCP/IP processor and engine using RDMA implementing a RDMA engine that provides various capabilities necessary for enabling remote direct memory access (*paragraph 0124 lines 13-15*) where the RDMA transfer may include read and write operation from a region with a certain number of bytes with a specific offset (*paragraph 0124 lines 80-86*).

Refer to claim 1 for motivational statement.

In regard to claim 37, Chung et al. teach the method of claim 35, further comprising: transmitting the memory access read command to retrieve previously unread portions of the

checkpoint data upon failure of the primary process (*Checkpoint server transmit last stored state to new primary application module, fig. 1, 110, H1-6, col. 4 lines 41-48*).

Chung et al. does not explicitly teach the method for transmitting the remote direct memory access.

Pandya teaches the TCP/IP processor and engine using RDMA implementing a RDMA engine that provides various capabilities necessary for enabling remote direct memory access (*paragraph 0124 lines 13-15*) where the RDMA transfer may include read and write operation from a region with a certain number of bytes with a specific offset (*paragraph 0124 lines 80-86*).

Refer to claim 1 for motivational statement.

\*\*\*\*\*

12. Claim 34 is rejected under 35 U.S.C. 103(a) as being unpatentable over Chung et al. (US 6,195,760) and in further view of Pandya (US 2004/0037319) in further view of St. Pierre et al. (US 6,141,773).

In regard to claim 34, Chung et al. and Pandya does not explicitly teach the method of claim 32, further comprising: appending updated checkpoint data to a previous set of the checkpoint data via a direct memory access write command.

St. Pierre et al. disclosed the method of backing up and restoring data in a computer storage system where differential backup is formed by the identified changed segments omitting at least on the segments that has not been changed (*col. 5 lines 30-63*).

A differential bit file may be constructed including copies of only the changed data segments (*fig. 13, 111a-111d*). The differential bit file captures changes to a logical entity as contiguous (*col. 17 lines 28-38*).

Refer to claim 14 for motivational statement.

\*\*\*\*\*

### ***Conclusion***

13. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. See PTO 892.

**THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to LOAN TRUONG whose telephone number is (571) 272-2572. The examiner can normally be reached on M-F from 8am-4pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, SCOTT BADERMAN can be reached on (571) 272-3644. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Loan Truong  
Patent Examiner  
Art Unit: 2114

/Gabriel L. Chu/  
Primary Examiner, Art Unit 2114